SYSTEM AND METHOD FOR REMOTE MANAGEMENT OF COMMUNICATIONS NETWORKS

FIELD OF INVENTION

[0001] The invention relates generally to the field of telecommunications. More specifically, but not by way of limitation, the invention relates to a system and method for remotely managing one or more communications networks.

BACKGROUND

[0002] Systems and methods are generally known for managing networks with the application of administrative consoles. One problem is how to enable network management by external service providers. A known solutions for management by an external service provider is to grant general administrative privileges to the service provider on a Local Area Network (LAN) or other network to be managed. Another known solution is to configure a firewall to permit access to the network from remote management consoles.

[0003] These known systems and methods for management by external service providers have several disadvantages. For example, direct connection to a LAN may not be feasible for a remote service provider. In addition, systems and method for modifying the configuration of a firewall may be costly to implement. Furthermore, approaches that result in broad administrative privileges to external service providers may present a security risk to stakeholders of data in the managed network.

[0004] What is needed is a system and method that facilitates remote management of one or more networks, while mitigating the risk associated with providing access through network firewalls.

SUMMARY OF THE INVENTION

[0005] The invention provides a system and method for that facilitating the remote management of one or more networks. In enabling the remote management of a network, embodiments of the invention provide limited access to service providers through a firewall, without the need to modify the configuration of the firewall. Advantageously, the cost of providing such access may be reduced compared to conventional approaches. In addition, such access may be limited to data inquiries or other commands, which can reduce the risk that the security of the network is compromised.

[0006] Embodiments of the invention provide a functional architecture having a control unit inside the firewall, and a proxy server outside the firewall. In one respect, embodiments of the invention provide a method to configure the control unit. In another respect, embodiments of the invention provide a method to configure the server. In yet another respect, embodiments of the invention provide a system and method for communicating between the control unit and the proxy server.

[0007] The features and advantages of the invention will become apparent from the following drawings and detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Embodiments of the invention are described with reference to the following drawings, wherein:

[0009] Fig. 1 is a block diagram of a functional architecture for a communications system, according to an embodiment of the invention;

[0010] Fig. 2 is a flow diagram of a process for remotely managing a network, according to an embodiment of the invention;

[0011] Fig. 3 is a flow diagram of a process for configuring a control unit, according to an embodiment of the invention;

[0012] Fig. 4 is a flow diagram of a process for configuring a proxy server, according to an embodiment of the invention;

[0013] Fig. 5 is a block diagram of a detailed functional architecture for a communications system, according to an embodiment of the invention; and

[0014] Fig. 6 is a flow diagram of a process for performing a communication session through a firewall, according to an embodiment of the invention.

DETAILED DESCRIPTION

[0015] Sub-headings are used below for organizational convenience. The disclosure of any particular feature is not necessarily limited to any particular section, however. The detailed description begins with an overview of a system architecture.

System Architecture

[0016] Fig. 1 is a block diagram of a functional architecture for a communications system, according to an embodiment of the invention. As shown therein, a proxy server 105 is coupled to console 120 and Internet 125 via a switch 110. Internet 125 is further coupled to firewall 135, which is coupled to console 130. In addition, Internet 125 is coupled to firewall 140, which is coupled to control unit 150 via LAN 145. Likewise, Internet 125 is also coupled to firewall 155, which is coupled to control unit 165 via LAN 160. Control units 150 and 165 may also be coupled to other networks (not shown) or devices (not shown).

[0017] As used herein, Internet 125 represents a public network. Internet 125 can be replaced with a Wide Area Network (WAN), Local Area Network (LAN), or other publicly-accessible wired or wireless network, according to design choice.

[0018] As used herein, LAN 145 and LAN 160 represent enterprise networks that are inside (i.e., on the protected side) of firewalls 140 and 155, respectively. LANs 145 and 160 could be or include WANs or other network configurations, according to application requirements.

[0019] Consoles 120 and 130 each may be or include a personal computer, a desktop computer, a laptop computer, a Personal Digital Assistant (PDA), or other terminal or device suitable for handling necessary user interface functions. Moreover, consoles 120 and 130 each may include client software to facilitate operation in a networked environment.

[0020] The control units 150 and 165 may each include a central processing unit (CPU) (not shown), such as an Intel x86, Intel x86 compatible device, Intel Pentium™, or other processor. The control units 150 and 165 may each further include a hard disk or other storage device (not shown) for storing programs and/or data. In addition, control units 150 and 165 may each have Random Access Memory (RAM), or other temporary memory (not shown) to execute Linux or other resident OS, and to execute application programs. Control units 150 and 165 may include application code (not shown) for managing LANs 145 and 160, respectively or for managing other networks (not shown) and/or devices (not shown). In addition, the control units 150 and 165 may each be or include a network server. In the illustrated embodiment, Control Units 150 and 165 are inside (i.e., on the protected side) of firewalls 140 and 155, respectively.

[0021] Proxy server 105 is a network-based server, and may include an Operating System (OS) (not shown), application code (not shown), and/or a database (not shown). In one respect, proxy server 105 provides access between each of consoles 120 and 130 and each of the control units 150 and 165. Control unit 150 may contain management data related to LAN 145 or other network (not shown) or network device (not shown), and control unit 165 may contain management data related to LAN 160 or other network (not shown) or network device (not shown). Proxy server 105 may aggregate and store performance data provided by control units 150 and 165, respectively. In operation, a user at either console 120 or 130 may provide commands to either or both of control units 150 and 165 via the proxy server 105.

[0022] In the illustrated embodiment, proxy server 105 is coupled to a public network. In an alternative embodiment, proxy server 105 may be protected inside a firewall (not shown). In yet another embodiment, proxy server 105 may be implemented within a De-Militarized Zone (DMZ) between a protected network (not shown) and the unprotected Internet 125 or other public network.

[0023] The quantity of any component illustrated in Fig. 1 may vary, according to application requirements.

Process Flows

[0024] Figs. 2- 4 illustrate enabling processes that can be performed using the functional architecture described above.

[0025] Fig. 2 is a flow diagram of a process for remotely managing a network, according to an embodiment of the invention. As shown therein, an overall process begins in step 205 by configuring a first or next control unit. Step 205 is described in more detail below, with reference to Fig. 3. Then, in conditional step 210, it is determined whether all control units have been configured. Where the result of conditional step 210 is in the negative, the process returns to step 205. If however, the result of conditional step 210 is in the affirmative, the process advances to step 215 to configure a proxy server. Step 215 is described in more detail below, with reference to Fig. 4. Finally, after both the control unit(s) and the proxy server have been configured, the process advances to step 220 to execute a communication session between the control unit(s) and the proxy server. Step 220 is described in more detail below, with reference to Figs. 5 and 6.

[0026] In an alternative embodiment of the process illustrated in Fig. 2, conditional step 210 is omitted. Thus, a communication session can be executed in step 220 after a single control unit is configured in step 205 and after the proxy server is configured in step 215.

[0027] Fig. 3 is a flow diagram of a process for configuring a control unit, according to an embodiment of the invention. The diagram is from the perspective of a control unit. As shown therein, the process begins in step 305 by receiving proxy server identification information. Such proxy server identification information may include, for example, server host name, IP address and logical port number.

[0028] Where a user, at console 120 or 130, for example, does not provide the server IP address, control unit 150 or 165 may obtain the server IP address using an inquiry command directed to the proxy server 105.

[0029] Next, the process advances to step 310 where the control unit 150 or 165 generates an access key. Finally, in step 315, the control unit 150 or 165 sends the access key and control unit identification information to the proxy server 105. Control unit identification information may include, for example, one or more of an external IP address and an internal IP address.

[0030] Accordingly, one or both of control units 150 and 165 are configured for remote communications with proxy server 105.

[0031] Fig. 4 is a flow diagram of a process for configuring a proxy server, according to an embodiment of the invention. The diagram is from the perspective of the proxy server 105. As shown therein, the process begins in step 405 by receiving control unit identification information from each of control units 150 and 165. Next, in step 410, the proxy server 105 stores the control unit information in a server database. Then, in step 415, the proxy server 105 adds each of control units 150 and 165 as remote devices. Finally, in step 420, a validation message may be exchanged between the proxy server 105 and each of the control units 150 and 165 to confirm the configuration of the control units and the server.

Communicating Through a Firewall

[0032] Fig. 5 is a block diagram of a detailed functional architecture for a communications system, according to an embodiment of the invention. As shown therein, a console 505 is coupled to a proxy server 510. The Proxy server 510 is coupled to a control unit 520 through a firewall 515. Proxy server 510 includes client request handler 525, shared request object pool 530 and server request handler 535. A request object 540 may be instantiated in any one or more of handler 525, pool 530, and handler 535.

[0033] In other embodiments, multiple consoles may be coupled to the proxy server 510. For example, consoles 120 and 130 could be substituted for console 505. In addition, in other embodiments, the proxy server 510 may be coupled to multiple control units through corresponding multiple firewalls. For instance, control units 150 and 165 could be substituted for control unit 520, and firewalls 140 and 155 could be substituted for firewall 515. Moreover, a proxy server 510 may have the features described above with reference to proxy server 105.

[0034] The operation of the functional components illustrated in Fig. 5, including messages 545, 550, 560, 565, 570 and 575 is described with reference to Fig. 6 below.

[0035] Fig. 6 is a flow diagram of a process for performing a communication session through a firewall, according to an embodiment of the invention. Fig. 6 is illustrated from the perspective of a proxy server. As shown therein, the process beings in step 605 by establishing a control port connection with a control unit. For example, proxy server 510 may establish a control port connection by receiving control port message 515 from the control unit 520 via firewall 515. The port connection may be used, for example, to open and close data connections, and/or to provide security functions.

[0036] Next, in step 610, the proxy server 510 establishes a connection with a console, and receives a request from the console. For instance, after establishing a server/client link with the console, which may be or include a Secure Socket Layer (SSL) link, proxy server 510 may receive console request message 550 from console 505.

[0037] Console request message 550 may be a request for network management data from control unit 520 related to LAN 145, LAN 160, other networks (not shown) coupled to control unit 150 and/or 165, or network devices (not shown) coupled to control units 150 and/or 165. For instance, console request message 550 may be a request for IP-PBX status information, where an IP-PBX is coupled to control unit 150 and/or 165. Console request message 550 may be a request for status information related to an Uninterruptible Power Supply (UPS) or other network device coupled to control unit 150 and/or 165. Further, console request message 550

may be a back-up, shut-down, re-start, or other control command directed to the control unit 150 and/or 165, or to an IP-PBX coupled to one of control unit 150 and/or 165, for example.

[0038] Then, in step 615, the proxy server 510 creates a request object having an identification (ID) number, where the request object is related to the console request. In addition, in step 615, the proxy server 510 adds the request object to a pool of one or more request objects. With reference to Fig. 5, step 615 may include creating request object 540 in client request handler 525. Step 615 may also include the assignment of ID number 0001 to request object 540, and the addition of request object 540 to object pool 530.

[0039] The process is then promoted to step 620, where the proxy server 510 notifies the control unit of a pending request object, by ID number. For example, proxy server 510 could send request pending message 555 to control unit 520 with notice of pending request object 540 having ID number 0001.

[0040] Next, in step 625, the proxy driver creates a data connection with the control unit, and receives a request from the control unit for a request object having a specific ID number. The data connection may be, for example, a TCP/IP socket, opened according to commands issued via the control port connection. As an illustration of the data flow over the data connection, proxy driver 510 could receive a get request message 560 from the control unit 520. For instance, the get request message could specifically request the request object 540 having ID number 0001.

[0041] Then, in step 630, the proxy driver retrieves the specified request object from the pool of one or more request objects. For example, with reference to Fig. 5, the server request handler 535 could retrieve request 540 having specified ID number 0001 from the shared request object pool 530.

[0042] The process then advances to step 635, where the proxy driver sends the specified request object to the control unit. Thus, in Fig. 5, the request handler 535 sends request 540 having the specified ID number 0001 to the control unit 520 as part of request message 565.

[0043] In step 640, the proxy server 510 receives a response to the specific request object from the control unit and closes the data connection with the control unit. For instance, in this step, the request handler 535 receives management data from the control unit 520 as part of control unit response message 570.

[0044] In step 645, the proxy server 510 sends the response to the console. As an example, request handler 535 could send the management data to the console 505 as part of proxy server response message 575. In this case, proxy server response message 575 contains management data from control unit 520 that satisfies console request message 550 from the console 505.

[0045] Finally, the proxy server 510 closes the connection with the console in step 650, for example by ending the SSL link between the console 505 and the proxy server 510.

[0046] The description above illustrates how the process in Fig. 6 can be executed by the functional architecture in Fig. 5. In addition, the process described with reference to Fig. 6 can be adapted to architectures having multiple consoles and/or multiple control units. To the extent that proxy server 510 includes at least one processor, the process illustrated in Fig. 6 may be embodied in processor-executable code, the processor-executable code being executed by the at least one processor.

CONCLUSION

[0047] The invention described above thus overcomes the disadvantages of known systems and methods by facilitating the remote management of one or more networks without requiring modification to a firewall protecting the network to be managed, and without granting broad administrative privileges to external service providers. While this invention has been described in various explanatory embodiments, other embodiments and variations can be effected by a person of ordinary skill in the art without departing from the scope of the invention.